

Title	Flood Filling Gameの計算量に関する研究
Author(s)	福井, 宏行
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/11317">http://hdl.handle.net/10119/11317</a>
Rights	
Description	Supervisor: 上原 隆平, 情報科学研究科, 修士

# On the computational complexity of the Flood Filling Game

Hiroiyuki Fukui (1010058)

School of Information Science,  
Japan Advanced Institute of Science and Technology

February 6, 2013

**Keywords:** Computational complexity, Flood Filling Game, Graph coloring, NP-complete, Polynomial time.

The Flood Filling games, which are called Flood-It!, Mad Virus, or Honey-Bee Game, are a kind of coloring games and they have been becoming popular online. In these games, each player colors one specified cell in his/her turn, and all connected neighbor cells of the same color are also colored by the color. This flooding or coloring spreads on the same color cells.

In this thesis, we deal with the following graph classes.

A graph  $(V, E)$  with  $V = \{v_1, v_2, \dots, v_n\}$  is an *interval graph* if there is a set of (real) intervals  $\mathcal{I} = \{I_{v_1}, I_{v_2}, \dots, I_{v_n}\}$  such that  $\{v_i, v_j\} \in E$  if and only if  $I_{v_i} \cap I_{v_j} \neq \emptyset$  for each  $i$  and  $j$  with  $1 \leq i, j \leq n$ . We call the set  $\mathcal{I}$  of intervals an *interval representation* of the graph. For each interval  $I$ , we denote by  $L(I)$  and  $R(I)$  the left and right endpoints of the interval, respectively (hence we have  $L(I) \leq R(I)$  and  $I = [L(I), R(I)]$ ). An interval representation is *proper* if no two distinct intervals  $I$  and  $J$  exist such that  $I$  properly contains  $J$  or vice versa. An interval graph is *proper* if it has a proper interval representation. If an interval graph  $G$  has an interval representation  $\mathcal{I}$  such that every interval in  $\mathcal{I}$  has the same length,  $G$  is said to be a *unit interval graph*.

A connected graph  $G = (V, E)$  is a *caterpillar* if  $V$  can be partitioned into  $B$  and  $H$  such that  $G[B]$  is a path, and every vertex in  $H$  is incident

to exactly one vertex in  $B$ . It is easy to see that the caterpillar  $G$  is an interval graph. We call  $B$  (and  $G[B]$ ) *backbone*, and each vertex in  $H$  *hair* of  $G$ , respectively.

A graph  $G = (V, E)$  is a *split graph* if  $V$  can be partitioned into  $C$  and  $I$  such that  $G[C]$  induces a clique and  $G[I]$  induces an independent set.

The Flood Filling Game is natural to consider the coloring games on general graphs. That is, once a vertex is colored, the flooding flows along edges in the graph. In the original Flood Filling Games, the player colors a specified cell. However, it is natural to allow the player to color any cell. The original game is called *fixed* and this extended game is called *free*. We consider the Flood Filling Games with or without an upper bound on the number of colors. We consider the optimization problem of finding the minimum solution to unify the color of all the cells of the board, and the deterministic problem of determining whether we can unify the color within  $t$  steps. We study how the computational complexity of these problems changes when the constraints like the number of colors or the graph classes change.

We define more precisely the generalization of this game. The game board is a connected, simple, loopless, undirected graph  $G = (V, E)$ . We denote by  $n$  and  $m$  the number of vertices and edges, respectively. There is a set  $C = \{1, 2, \dots, k\}$  of colors, and every vertex  $v \in V$  is precolored (as input) with some color  $col(v) \in C$ . For a vertex set  $U \subseteq V$ , the vertex induced graph  $G[U]$  is the graph  $(U, F)$  with  $F = E \cap \{\{u, v\} \mid u, v \in U\}$ . For a color  $c \in C$ , the subset  $V_c$  contains all vertices in  $V$  of color  $c$ . For a vertex  $v \in V$  and color  $c \in C$ , we define the *color- $c$ -neighborhood*  $N_c(v)$  by the set of vertices in the same connected component as  $v$  in  $G[V_c]$ . For a given graph  $G = (V, E)$  and a sequence  $(v_1, c_1), (v_2, c_2), \dots, (v_t, c_t)$  of coloring operations in  $V \times C$ , we let  $G_0 = G$  and  $G_i$  is the graph obtained by the coloring operation  $(v_i, c_i)$  on  $G_{i-1}$  for each  $i = 1, 2, \dots, t$ . Recently, computational complexities of the variants of the Flood Filling Games on several graph classes have been studied.

In this thesis, we investigate the Flood Filling Games on some graph classes, and obtain the following results.

- The Free Flood Filling Game is NP-complete even on trees of three colors.

- The Free Flood Filling Game on a path/cycle can be solved in  $O(n^3)$  time and  $O(n^2)$  space.
- The Free Flood Filling Game is **NP**-complete even on caterpillars when the number of colors is not bounded.
- The Free Flood Filling Game is **NP**-complete even on proper interval graphs when the number of colors is not bounded.
- The Free Flood Filling Game is **NP**-complete even on split graphs when the number of colors is not bounded.
- The Free Flood Filling Game on a split graph of  $k$  or fewer colors can be solved in  $O((k!)^2 + n)$  time.

Our results state that the number of colors is a key parameter to determine the computational complexity of the Flood Filling Games. If the number of colors is not bounded, the Free Flood Filling Game is **NP**-complete even on caterpillars, proper interval graphs, and split graphs. Both of the classes of caterpillars and proper interval graphs consist of very simple interval graphs. If the maximum degree is bounded by 2, these classes degenerate to the set of paths. Thus the results are tight. On the other hand, when the number of colors is a fixed constant, the game can be solved in polynomial time on split graphs.